Cross-layer Visibility as a Service

Ramana Rao Kompella[*], *Albert Greenberg*[†], *Jennifer Rexford*[‡], *Alex C. Snoeren*[*], *Jennifer Yates*[†]
[*]*UC San Diego* [†]*AT&T Labs–Research* [‡]*Princeton University*

Accurate cross-layer associations play an essential role in today's network management tasks such as backbone planning, maintenance, and failure diagnosis. Current techniques for manually maintaining these associations are complex, tedious, and error-prone. One possible approach is to widen the interfaces between layers to support auto discovery. We argue instead that it is less useful to export additional data between layers than to import information into a separate, logically centralized management database. The specification of an interface to this database enables independent evolution of individual layers, side-stepping the challenges inherent in wide layer interfaces. Furthermore, management tools can leverage the network-wide cross-layer visibility provided by such a database to deliver enhanced services that depend on physical- or link-layer diversity.

1 Introduction

The Internet continues to lack the level of reliability and robustness we expect from critical infrastructure. Network events, such as equipment failures and planned maintenance, often cause disruptions in service or even loss of connectivity. Diagnosing the root cause of failures is important, yet surprisingly difficult. While traditional layering in IP networks helps contain complexity within simple abstractions, we argue that *poor visibility across layers* is a major impediment to the manageability and in turn reliability of the network.

In essence, a link at one layer (e.g., IP) consists of a path—a sequence of components—at the next layer (e.g., fibers and optical amplifiers). Greater visibility across layers would significantly improve network planning, risk assessment, fault diagnosis, and network maintenance. In practice, ISPs address these problems by maintaining complex databases and analyzing large amounts of topology, configuration, and measurement data collected from network elements at each layer. This approach is driven primarily by the absence of any immediately viable alternative, since most layers have little or no visibility into the other layers. As a long-term solution to this seemingly ad hoc approach, we could imagine "fattening" the interface between layers to make network elements more aware of dependencies they inherit from the layers below and impose on the layers above. These fat interfaces would enable network elements to select diverse paths that avoid shared risks and provide greater visibility to troubleshooting tools like traceroute.

Despite the apparent advantages of wider interfaces, we argue that this is the wrong approach to the problem, *even if we had the luxury of a clean-slate redesign of the interfaces between layers.* First, the simple abstraction of a link plays an important role in containing complexity inside the network; wider interfaces make it harder for each layer to evolve independently. Second, fundamental reasons prevent some network elements from easily providing information about shared risks to the layers above them; for example, a fiber cannot easily notify an IP link about the underground locations it traverses, and whether other fibers lie nearby. Third, having the network elements store historical data and answer queries is challenging, because of the overhead involved and the need for providers to control access to the data for business and security reasons. Finally, and perhaps most importantly, detailed cross-layer visibility is essential to designing, managing, configuring, and troubleshooting the network, making it appealing to store the information *outside* of the network elements.

Yet today's ad hoc approach of collecting and analyzing data in home-grown databases is not a sufficient solution. Instead, we argue that cross-layer visibility should be provided as a *service*, with well-defined interfaces for populating the external databases and querying the information. Rather than just dictating what the network elements store and export—the approach taken by the Simple Network Management Protocol (SNMP) [2]—we focus on what information is *imported* into the management database. This subtle distinction is extremely important, as it allows many different solutions for providing the information. Although the network elements themselves may generate the data, information could also come from separate measurement devices or even human operators. This approach accommodates the inherent diversity across layers and the natural evolution of techniques for collecting the data. We present a possible evolution path for four layers: identifying fibers running through the same geographic location, mapping an IP link to optical components, determining the IP forwarding path, and identifying the intermediate hops in an overlay path.

Greater uniformity in the data representation makes it easier to evolve a network, integrate two networks after an acquisition, and employ third-party network-management tools. More broadly, we argue that the management system should have interfaces for different stake holders—such as network designers, network managers, and customers—to query the data, with explicit policies governing the kinds of information each party can access. For example, a customer, using such a system, could ask if two IP paths (or two access links) are logically diverse, but the provider might choose to reveal such information depending on the customer. In contrast, the system could allow a network manager troubleshooting a reachability problem to perform a complete "traceroute" across all of the layers. A network designer could conduct a "what-if" analysis of the effects of planned maintenance on the link loads. The system can also keep a log of past queries, to learn more about the cause and impact of failures by analyzing patterns in the queries.

2 The case for cross-layer visibility

Layering in IP networks fundamentally hides the complexity of lower (upper) layers and exposes very simple interfaces to upper (lower) layers. This allows parallel and independent evolution of layers while still preserving the interface between them. However, strict layering, as in the current network architecture, results in *poor visibility* across layers affecting certain operational tasks that rely on accurate cross-layer visibility.

In today's IP backbone networks, each IP link consists of a connected set of optical components organized in different topologies (e.g., ring, mesh, etc.). A single link consists of many different optical components and multiple links can share a particular component, thus creating a many-to-one, one-to-many mapping. Cross-layer visibility refers to the associations between higher layer abstractions to lower layers and vice versa. For example, cross-layer visibility in IP networks refers to the association between an IP point-to-point link and the set of optical components that comprise the link.

Accurate associations are critical to the functioning of various operational tasks, including the following:

- *Backbone planning.* Backbone planning involves engineering the network to withstand a wide range of potential failure scenarios, planning traffic growth, avoiding single points of failures (Shared Risk Link Groups [14]) and supporting additional services and features in the network. An accurate audit of the network that transcends all layers, therefore, is a key ingredient in backbone planning.
- *Customer fault-tolerance.* Customers typically obtain diversity in their network connectivity either by multi-homing to two different points-of-presence (PoPs) within the same provider, or to two different carriers. One common question they face is whether there are any shared risks lurking (e.g., unprotected circuits on fibers passing through the same tunnel). Necessarily, customers need accurate cross-layer mappings to ensure this diversity. Of course, it is a matter of the ISP's policy whether to disclose such information to customers.

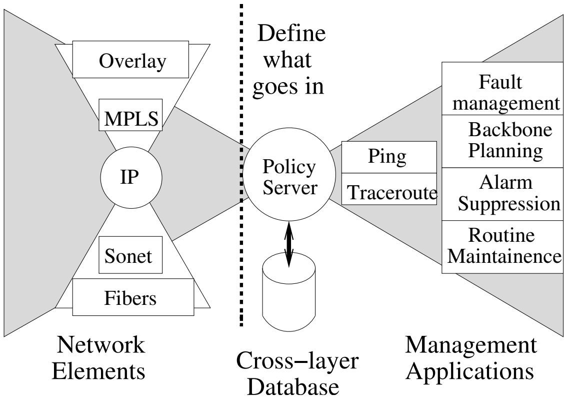

Figure 1: A bow-tie architecture to provide cross-layer visibility as a service to management applications.

- *Maintenance.* Network operators often gracefully remove the traffic on a link (e.g., by increasing the link's OSPF weight) before performing various maintenance activities such as repairing a faulty component, provisioning a new link, software upgrades and so on. Besides, operators usually perform a "what-if" analysis before maintenance to ensure smooth functioning of the network. Maintenance operations can induce unwanted faults into the network if upper layer names (e.g., IP links) are misassociated with lower layer names (e.g., SONET circuits) or vice versa.

On the surface, it seems that maintaining accurate associations should be a straightforward task. After all, network operators provision their network in a centralized manner; therefore, they can log these associations in databases. However, a live operational network incurs significant churn as links are provisioned, old equipment is replaced, faulty components are repaired, interfaces are re-homed and so on. Database errors can result from this inherent churn—for example, operations may fail to update the relevant databases as an IP link is moved from a failed line card to a different, operational card.

Additionally, during failures, there are topological changes at some (or even all) layers that can lead to considerable shifts in topology. An example of such a change is observed when the IP layer reroutes traffic through alternate paths during link failures. Corresponding restoration at lower layers (e.g., optics) retains IP level logical topology, but changes the optical light paths. These changes in topology make it hard to diagnose failures or other management tasks without the presence of accurate and up-to-date cross-layer associations.

3 Architecture for cross-layer visibility

We propose a "bow-tie" architecture for providing cross-layer visibility that is analogous to the "hour-glass" architecture of IP that facilitated the rapid evolution of the Internet. In our architecture, shown in Figure 1, a network-centric view of the layered IP stack forms the left half of the bow-tie. (Conceptually, MPLS-on-IP and application-layer overlays are in-network analogs to end hosts' transport and application layers.) The center is a cross-layer database that provides service to the network-management applications that form the right half of the bow-tie.

3.1 Bow-tie architecture

IP networks are constructed and thereby managed in a layered fashion, with a relatively thin interface between layers. This approach has enabled rapid deployment of new technologies on the Internet as changes in devices that belong to a particular layer are isolated from layers above or below. This allows parallel and independent evolution of all the layers. In our architecture, we do not propose changes to the layering in the current hourglass model. Instead, we outline an evolution strategy for devices in each of these layers that will greatly enhance the accuracy of the cross-layer associations stored in a separate network-management database.

At the center of the bow-tie is a centralized cross-layer policy server that interfaces between the network elements, the cross-layer database, and the various management applications that are built on top of the server. The policy server interacts with various network elements at different layers using a combination of standard mechanisms (e.g., SNMP) and non-standard third-party mechanisms (e.g., OSPF monitor [11]) to populate the cross-layer database. The policy server supports both archiving of data as well as on-demand fetching of information from the network, thus controlling the "liveness" of the data reported to the various management applications. Finally, the policy server also controls what information can be exported to various users (such as end-users, ISP customers and ISP operations) based on the particular provider's policy.

We attach a cross-layer database (possibly distributed) to the policy server that stores the topology at each layer and mappings from elements at one layer to the supporting set of components at the previous layer. For example, the database stores the IP topology (i.e., the routers and the links between them) as well as the forwarding paths between each pair of routers. The database also stores the optical topology and indicates what sequence of optical components (such as fibers and amplifiers) form each IP link between adjacent IP routers. Similarly, the database keeps track of what fibers run through the same conduit, as well as the geographic path the conduit traverses from one termination point to another. The database has unique names for devices at each layer, as well as indices necessary to map between layers. Of course, the accuracy of this cross-layer database depends entirely upon the accuracy of the information exported by individual network elements.

Finally, at the right end of the bow-tie are the management applications that are built on top of the cross-layer database. Exporting potentially different, policy-controlled views to each applications allows for the construction of a scalable management architecture. Section 4 describes several example applications that can be easily built on top of this cross-layer database. In particular, we show how automated fault diagnosis and mitigation, automatic reconfiguration, and other tasks suggested as motivating applications of the Knowledge plane [3] benefit from the presence of our cross-layer database.

3.2 Comparison with current practices

One may ask "In what way is our proposal any different from current practice?" Today's management databases are a mixture of human-generated inventory and measurement data, with little compatibility from one Autonomous System (AS) to another; even within a single AS, the representation of data often changes over time as the network design and measurement infrastructure evolve. The poor level of uniformity makes it exceptionally difficult to evolve a network, integrate two networks after an acquisition, or incorporate third-party network management tools.

We believe that part of the problem is that the research and standards community has focused only on defining the information that individual network elements export (e.g., SNMP Management Information Bases and NetFlow measurement records). Additionally, there is an acute need to standardize what a management database imports (e.g., IP topologies and traffic matrices). By standardizing the data models a database imports, vendors will be able to independently choose an appropriate implementation while preserving interoperability.

Often, the views needed by network-management applications are not available from any one network element, and must be constructed by joining data from many parts of the network. In addition, there are multiple viable ways to construct these views, depending on the sophistication of the network elements and the monitoring infrastructure. Specifying what goes *into* the database allows network technologies and monitoring infrastructure to gradually evolve over time improving accuracy while preserving the interface with the management systems.

3.3 Advantages of the bow-tie architecture

Providing cross-layer visibility to various applications as a service has several advantages. First, it results in lower overhead for the routers; queries are redirected to the management system where appropriate information (based on policy) can be generated, rather than implementing the policies in the routers themselves. The system can also cache the results of recent or common queries to reduce the overhead of satisfying future queries. This sharing of results through a common cache across applications makes the mechanism scalable, allaying concerns of over-burdening network elements with management tasks.

By maintaining a log of network changes over time, the service can answer queries that require historical data. For example, a customer could inquire about a performance problem that started ten minutes ago, and the service could report whether a failure forced the customer's traffic onto a path with a longer round-trip time. The management system can also implement explicit policies to control what kind of information is revealed, and to whom. For example, a customer may be allowed to ask if two paths have a shared risk, but not learn exactly what component is shared and where it is located. By forcing all queries through the service, the AS can protect its routers from probe traffic while still providing good network visibility to customers. In addition, the very notion of a shared risk is extremely subjective [14], and the service can accommodate this by allowing queries at different granularity and incorporating extra information. For example, a network designer may want to know if two fibers lie near the San Andreas Fault in San Francisco. Or, one customer might be interested in link-disjoint paths and another in PoP-disjoint paths through the network.

With a common database interface, ASes could cooperate to provide greater visibility into shared resources. For example, an ISP that leases fiber from another provider could automatically learn the geographic path it follows (abstracted as deemed fit by the providers), or a multi-homed customer could determine its vulnerability to failures affecting both of its providers. Or, a governmental agency could conduct a realistic study of the effects of a serious catastrophe (such as a terrorist attack) on the Internet infrastructure.

3.4 Alternate approaches

One approach to obtaining better cross-layer visibility is to widen the interfaces between layers. A rich interface between layers allows individual layers to gain insight into the behavior of adjoining layers. For example, if the network layer (IP/MPLS) were made aware of the underlying components in optical topology, the network layer may be able to make better choices in recovering from failure situations. Indeed, in the context of fast restoration from failures in the MPLS domain, Interior Gateway Protocol (IGP) extensions [6, 13] incorporate shared risk link groups (SRLGs) in their link state advertisements (LSAs). The SRLGs themselves could be populated through management plane through manual or semi-automatic means. For example, the link management protocol (LMP) [4] allows automatic identification of neighbors in the optical domain. Therefore, all the optical components associated with a link could be identified by tracing the neighbors from one router to another (that constitute the link). However, LMP does not identify physical shared risks automatically; sections of fibers common between two IP links are not automatically identified by LMP. While such techniques encourage new proposals to expose lower layers to upper layers in order to obtain cross-layer visibility, we argue that this approach does not scale well.

First, exposing detailed lower-layer topology to upper layers adds complexity into the network (increased processing due to new types of messages) and limits scalability (too many devices results in higher messaging overhead). Second, interoperability is difficult to ensure across different types of devices; the larger the number of devices that need to be interoperable, the more difficult it becomes to achieve consensus on one protocol. Besides, it necessitates long design and testing cycles across large number of devices and manufacturers. Third, wide interfaces can impact the security of the network; the compromise of one network element (either physically tapping a fiber or through network exploits) could expose the details of a large portion of network—including both lower and higher layers. Finally, it is fundamentally difficult to achieve complete cross-layer visibility (e.g., automatically identifying proximity of two fibers, or two geographical properties such as fault lines, etc) in this fashion.

One could also argue that automatic reconfiguration and self-healing within layers completely obviates the need for cross-layer visibility. For example, if a SONET ring automatically recovers from a failure by re-routing the traffic the other way around the ring, there is no obvious need to inform the higher layers. In "intelligent" optical networks, optical-layer restoration causes light paths to automatically re-route from primary to alternate paths to avoid faults. These dynamic path changes at lower layers are typically achieved without impacting the upper layer connectivity; IP links are, by design, oblivious to restoration at lower layers.

We argue that, while IP links can be oblivious to lower-layer restoration and in fact should be, the management system itself still requires cross-layer visibility. Of course, restoration at lower layers (such as optical re-routing) is far more expensive than IP-level restoration; thus optical layer protection is often not used—particularly on high speed links [8]. Besides the cost issue, optical reroutes may result in subtle changes in IP-layer performance metrics such as end-to-end delay that are important for certain applications. Reliable diagnoses of these performance anomalies and other faults across layers require an accurate cross-layer view of the network.

4 Applications

We now present a set of concrete example applications that can benefit from our architecture.

4.1 Cross-layer traceroute

Traceroute is one of the most common diagnostic tools used to identify the IP forwarding path from a given source to a destination. Providers would like the fidelity of information reported by traceroute to vary depending on the user who initiates it. For example, probes from an end user might not reveal failures at lower-layer network elements (e.g., individual switches within a provider). Large enterprise networks often use traceroute to diagnose whether a given problem lies within their network or a provider's network and, as such, the provider can choose to expose more information to the enterprise than an end-user. Finally, a network operator would like to obtain full topology information for the entire network; as such, probes originated by the operator obtain unrestricted access to information.

In the current architecture, policies are enforced in a distributed ad hoc fashion throughout the network, placing additional burden on individual network elements. Instead, in our proposed architecture, all traceroute queries can be directed to the cross-layer database, where appropriate information can be exposed depending on who initiated the request, and new measurements taken if necessary to provide a more timely or more accurate response.

4.2 Backbone planning and maintenance

Various backbone planning and maintenance tools rely extensively on accurate cross-layer associations. The first step in network maintenance often involves gracefully rerouting the traffic on a given link (by increasing the OSPF weight of a link or some such mechanism) before repairing a faulty component, provisioning a new link, installing software upgrades, etc.; mis-associations across layers could lead to the decommissioning of the wrong link. During failures, a large flurry of alarms at different layers are generated by various network elements. An accurate diagnosis (either manual or through automated correlation tools [5, 7, 12]) requires grouping these alarms together based upon cross-layer associations.

In today's ISPs, these associations are typically generated either by hand through careful merging of router configurations. In our architecture, through standardization of the data model for importing these associations, we can incorporate various manual as well as automated mechanisms to obtain cross-layer associations, even as the mechanisms themselves evolve over time.

4.3 Customer fault-tolerance

Customers (e.g., e-commerce businesses) are primarily interested in obtaining uninterrupted network connectivity either from one single service provider or through different service providers via multi-homing. One common challenge they face is ensuring diversity in their connectivity to the Internet backbone.

In our architecture, these questions can be answered very easily: the customer can initiate queries to the management system that consults the cross-layer database to obtain accurate mappings. Of course, whether to disclose physical connectivity information—either completely or in part—is often a policy decision that can be implemented easily through the cross-layer policy-server and database.

5 Evolution path

The ultimate accuracy of the cross-layer associations is limited by both technological as well as cost issues. However, by defining the data *imported* by the management system, rather than *exported* by the network elements, our architecture supports many ways of learning the intra-layer topology and paths, and the associated cross-layer mappings.

5.1 Fiber and fiber spans

A fiber map captures the topology of the underlying transport network. A fiber consists of multiple *spans*, a segment of fiber traversing a single conduit. A fiber span, in turn, consists of multiple fibers traversing the same conduit. This information could be learned in various ways. As with other optical components, the operators can keep track of the location of fiber and the mapping to/from spans *manually* as the fibers are installed, or leased from other providers. In addition, one could deploy *intelligent fibers and conduits* to automatically advertise their operational status (e.g., Loss of Signal). Creating new techniques for auditing the management database, or even automatically generating the data, is an exciting direction for future research. For example, optical amplifiers along the optical path could report their identity and geographic location [10]. In addition, the individual fibers could have RFID tags where they enter and leave a conduit. For even higher accuracy, the conduits could have active devices, such as audio or wireless transmitters coupled with GPS receivers, to be placed at fixed intervals. Alternately, to verify the mapping of fibers to IP links, we could envision injecting labels or tags into fibers and periodically tapping the fiber to report these labels along with their positioning using a GPS. Such sophisticated mechanisms can be incrementally deployed in the network to obtain fine-grained associations.

5.2 Optical components and paths

The optical topology consists of a diverse array of devices, including fibers, amplifiers, cross connects, and add-drop multiplexers. The sequence of optical components underlying an IP link could be learned in various ways, depending on the sophistication of the optical components. Of course, the operators can keep track of optical components and their relationships to IP links *manually* as the equipment is installed and apply heuristics [5, 7] to perform consistency checks. The manual approach, however, does not suffice in the presence of lower-layer restoration as the path can change dynamically. Capturing these changes requires logging of alarms or *periodic probing* of the adaptive components and correlation across layers. Discovering the optical components that comprise a link becomes much easier with neighbor discovery mechanisms such as LMP. Of course, even within the the optical components there are multiple layers. Each layer could employ different mechanisms to *automatically* obtain different subsets of the path, that could then be then joined in an offline fashion. For example, a combination of an LMP like protocol in regular optical networks together with configuration state in intelligent optical networks (with dynamic restoration capabilities) could be used to track the components along different sections of the path.

5.3 IP topology and forwarding paths

The IP-level topology for an AS consists of routers and links, and a forwarding path consists of one or more sequences of IP links. The topology and paths can be learned in various ways, with different degrees of accuracy and timeliness. The topology can be recorded in a *static* manner by the operators as equipment is installed, or reverse-engineered from the router configuration state. Alternately, a monitoring system can *periodically* poll the routers for their status and forwarding tables, or run traceroute probes to map the topology. Finally, a monitor could collect routing-protocol messages, field alarms when equipment goes up/down, or analyze syslog output generated by the routers to provide an up-to-date view of the topology and paths.

5.4 MPLS and overlay paths

MPLS allows label-switch paths (LSPs) between two routers (typically the ingress and egress routers of an AS) to be configured either statically or dynamically. In the static case, the path is known during configuration itself, while in the dynamic case, a path is selected based on the IP forwarding state. An overlay path between two nodes could be a combination of MPLS enabled as well as regular IP-forwarded paths through intermediate hosts. A cross-layer service could keep track of the sequence of intermediate nodes, or LSP end points, along an end-to-end path between two hosts. These associations could be determined by monitoring the signaling messages used to establish these paths, or by receiving reports from the routers or overlay nodes.

6 Conclusion

This paper addresses the challenges of providing cross-layer visibility to network-management applications, and advocates against expanding the interfaces between layers for auto-discovery of the cross-layer associations. Instead, we propose an architecture where such associations can be learned or maintained automatically, not by widening the layers, but by defining the data that should be imported into a management database. The architecture provides cross-layer visibility as a service to other applications and users that depend on this information.

7 Acknowledgments

We wish to thank the anonymous reviewers for their valuable comments that greatly helped improve this paper. The first author is supported in part by NSF grant ANI0074004.

References

[1] Update: CSX train derailment, thread on NANOG mailing list, `http://www.merit.edu/mail.archives/nanog/2001-07/msg00367.html`, July 2001.

[2] J. Case, M. Fedor, M. Schoffstall, and J. Davin. A simple network management protocol (SNMP). RFC 1157, Internet Engineering Task Force, May 1990.

[3] D. D. Clark, C. Partridge, J. C. Ramming, and J. T. Wroclawski. A knowledge plane for the internet. In *ACM SIGCOMM*, 2003.

[4] A. Fredette and J. Lang. Link management protocol (LMP) for DWDM optical line systems. RFC 4209, Internet Engineering Task Force, Oct. 2005.

[5] S. Kandula, D. Katabi, and J. P. Vasseur. Shrink: A tool for failure diagnosis in IP networks. In *Proc. ACM SIGCOMM MineNet Workshop*, Aug. 2005.

[6] D. Katz, K. Kompella, and D. Yeung. Traffic engineering extensions to OSPF version 2. RFC 3630, Sept. 2003.

[7] R. Kompella, J. Yates, A. Greenberg, and A. C. Snoeren. IP fault localization via risk modeling. In *Proc. Networked Systems Design and Implementation*, May 2005.

[8] G. Li, D. Wang, R. Doverspike, C. Kalmanek, and J. Yates. Economic analysis of IP/Optical network architectures. In *Proc. Optical Fiber Communication Conference*, Mar. 2004.

[9] E. Mannie. Generalized multi-protocol label switching (GMPLS) architecture. RFC 3945, Oct. 2004.

[10] P. Sebos, J. Yates, D. Rubenstein, and A. Greenberg. Effectiveness of shared risk link group auto-discovery in optical networks. In *Proc. Optical Fiber Communication Conference*, Mar. 2002.

[11] A. Shaikh and A. Greenberg. OSPF monitoring: Architecture, design and deployment experience. In *Proc. USENIX Symposium on Networked System Design and Implementation (NSDI)*, Mar. 2004.

[12] SMARTS Inc. `http://www.smarts.com`.

[13] H. Smit and T. Li. Intermediate system to intermediate system (IS-IS) extensions for traffic engineering (TE). RFC 3784, June 2004.

[14] J. Strand, A. Chiu, and R. Tkach. Issues for routing in the optical layer. In *IEEE Communications Magazine*, Feb. 2001.